NATIONAL POLAR-ORBITING OPERATIONAL ENVIRONMENTAL SATELLITE SYSTEM (NPOESS)

OPERATIONAL ALGORITHM DESCRIPTION DOCUMENT FOR VIIRS CLOUD TOP PARAMETERS (CTP) EDR SOFTWARE (D39568 Rev B)

CDRL No. A032

Northrop Grumman Space & Mission Systems Corporation
One Space Park
Redondo Beach, California 90278

This document has been identified per the NPOESS Common Data Format Control Book – External Volume 5 Metadata, D34862-05, Appendix B as a document to be provided to the NOAA Comprehensive Large Array-data Stewardship System (CLASS) via the delivery of NPOESS Document Release Packages to CLASS.

NATIONAL POLAR-ORBITING OPERATIONAL ENVIRONMENTAL SATELLITE SYSTEM (NPOESS)

OPERATIONAL ALGORITHM DESCRIPTION DOCUMENT FOR VIIRS CLOUD TOP PARAMETERS (CTP) EDR SOFTWARE (D39568 Rev B)

ELECTRONIC APPROVAL SIGNATURES:

Roy Tsugawa Date
Algorithm & Data Processing IPT Lead &
Algorithm Change Control Board Chairperson

Ben James Date
Operations & Support IPT Lead

The following individuals are recognized for their contributions to the current or previous versions of this document.

David Hogan
Eric Wong
Sung Joo

Revision/Change Record

Document Number **D39568**

Revision	Document Date	Revision/Change Description	Pages Affected
---	9-27-04	Initial Release.	All
A	10-5-04	Minor typographical corrections, correct page numbering, correct several table references.	Sections 2.1, 6.3, App. B
B1	10-29-04	Revise Tables 22-27, correct page number Per SPCR ALG 647.	All
B2	10-29-04	Revisions based primarily on comments and suggestions from IDPS review (R. Slonaker 20 Oct 2004) and suggestion by the SE M&S Group. Key changes are (SPCR ALG 647): expanded description of functions (input, output arguments) revised/ corrected diagrams minor corrections and clarifications based on IDPS comments.	All

Revision/Change Record

Document Number D39568

Revision	Document Date	Revision/Change Description	Pages Affected
B3	4-19-07	20Sep05 - Reflects Raytheon-Omaha's initial edits for Science To Operational (Sci2Ops) Code Conversion Process, adding Raytheon coversheet, title/signature page, etc. 18Jan06 – Made minor edits per the 11Jan06 CUTPR, fixing TBD/TBR table, some font inconsistencies, etc. 06 & 07Mar06 - Reflects minor changes to address comments from CTP EH & CSI OPT CUTPR 20060303. 20Mar06 – Edits done to capture Sci2Ops code conversion process. 18Apr06 – Edits done to capture Sci2Ops code conversion process. 2May06 – Edits done to capture Sci2Ops code conversion process fixing TBD/TBR table. 8May06 – Added Unit Test dated 11Apr06 posted for CCPR into Section 5.1 per the way OAD content has been done the last couple of years until a new OAD template says to remove this information. Updated page numbers for Table of Contents, List of Tables, List of Figures. Changed List of Tables and List of Figures from Times New Roman bold to Arial, 11, non-bold to match the Table of Contents fonts. Made other minor edits. 11May06 – Made final minor edits and sent document to NPOESS_CM for official delivery to NGST. 19Apr07 – Editorial change. Changed the Ancillary Inputs table to note that we actually input meters for Height_Profile and Tropopause_Height data and convert those data sets into kilometers. This did not change in the code, we are just adding the note to the table for clarification. The note is already in the paragraph above the Ancillary Inputs table. Changes made according to NP-EMD.2006.510.0029 and NP-EMD.2006.510.0093.	All
B4	5-21-07	Delivered to NGST.	All
B5	2-14-08	Added hPa as units for surface pressure.	All
B6	6-19-08	Reformatted to new OAD template. Updated in response to NGST comments from last delivery. TM NP-EMD.2005.510.0079 has been incorporated into CTP. Delivered to NGST.	All
B7	7-11-08	TM NP-EMD-2008.510.0004 has been incorporated into CTP.	All
B8	10-1-08	Updated Graceful Degradation. Prepared for TIM/ACCB.	All
B9	10-21-08	TM NP-EMD-2008.510.0048 has been incorporated into CTP.	All

	NPOESS Northrop Grumman Raytheon DOC • DoD NASA

Revision/Change Record

Document Number D39568

Revision	Document Date	Revision/Change Description	Pages Affected
B	1-14-09	ECR A-172 and A-195. Incorporated interim changes and addressed TIM comments. Corrected Greek letter mu/box issue. Corrected Reference Source Error. Approved for Public Release per Contracts Letter 100610-02.	All

Table Of Contents

List of Figures

List of Tables

1.0 INTRODUCTION

1.1 Objective

The purpose of the Operational Algorithm Description (OAD) document is to express, in computer-science terms, the remote sensing algorithms that produce the National Polar-Orbiting Operational Environmental Satellite System (NPOESS) end-user data products. These products are individually known as Raw Data Records (RDRs), Temperature Data Records (TDRs), Sensor Data Records (SDRs) and Environmental Data Records (EDRs). In addition, any Intermediate Products (IPs) produced in the process are also described in the OAD.

The science basis of an algorithm is described in a corresponding Algorithm Theoretical Basis Document (ATBD). The OAD provides a software description of that science as implemented in the operational ground system --- the Data Processing Element (DPE).

The purpose of an OAD is two-fold:

1. Provide initial implementation design guidance to the operational software developer
2. Capture the “as-built” operational implementation of the algorithm reflecting any changes needed to meet operational performance/design requirements

An individual OAD document describes one or more algorithms used in the production of one or more data products. There is a general, but not strict, one-to-one correspondence between OAD and ATBD documents.

This particular document describes operational software implementation for the Visible/Infrared Imager/Radiometer Suite (VIIRS) Cloud Top Parameters (CTP) Environmental Data Record (EDR).

1.2 Scope

The scope of this document is limited to the description of the core operational algorithm(s) required to create the VIIRS CTP EDR software. The theoretical basis for this algorithm is described in Section 3.3 of the Cloud Top Parameters ATBD, D43754.

1.3 References

The primary software detailed design documents listed here include science software documents, NPOESS program documents, plus source code and test data references.

1.3.1 Document References

The science and system engineering documents relevant to the algorithms described in this OAD are listed in Table 1.

Table 1. Reference Documents

Document Title	Document Number/Revision	Revision Date
Cloud Top Parameters Algorithm Theoretical Basis Document (ATBD)	D43754 Rev. A	28 Nov 2007
VIIRS Cloud Top Parameters Unit Level Detailed Design	P1187-SW-I-007 Ver. 6 Rev. 5	16 Jun 2005
VIIRS Radiometric Calibration Unit Level Detailed Design	Y2490 Ver. 5, Rev. 4	30 Sep 2004
Operational Algorithm Description Document for the VIIRS Cloud Mask Intermediate Product (VCM IP) Software	D36816 Rev. A	3 Dec 2004
NPP EDR Production Report	D37005 Rev. C	16 Mar 2007
EDR Interdependency Report	D36385 Rev. C	7 Nov 2007
CDFCB-X Volume I - Overview	D34862-01 Rev. B	27 Aug 2007

Document Title	Document Number/Revision	Revision Date
CDFCB-X Volume II – RDR Formats	D34862-02 Rev. B	27 Aug 2007
CDFCB-X Volume III – SDR/TDR Formats	D34862-03 Rev. A	27 Aug 2007
CDFCB-X Volume IV Part 1 – IP/ARP/GEO Formats	D34862-04-01 Rev. A	10 Sep 2007
CDFCB-X Volume IV Part 2 – Atmospheric, Clouds, and Imagery EDRs	D34862-04-02 Rev. A	10 Sep 2007
CDFCB-X Volume IV Part 3 – Land and Ocean/Water EDRs	D34862-04-03 Rev. A	10 Sep 2007
CDFCB-X Volume IV Part 4 – Earth Radiation Budget EDRs	D34862-04-04 Rev. A	10 Sep 2007
CDFCB-X Volume V - Metadata	D34862-05 Rev. B	27 Aug 2007
CDFCB-X Volume VI – Ancillary Data, Auxiliary Data, Reports, and Messages	D34862-06 Rev. C	10 Sep 2007
CDFCB-X Volume VII – Application Packets	D34862-07 Rev. ---	10 Sep 2007
NPP Mission Data Format Control Book (MDFCB)	GSFC 429-05-02-42 R1	14 Apr 2006
NPP Command and Telemetry (C&T) Handbook	568423 Rev. A	5 Apr 2005
NGST/SE technical memo – MS Engineering Memo_CTP OAD Update	NP-EMD.2005.510.0079 Rev. ---	7 Jul 2005
NGST/SE technical memo – NPP_VIIRS_CTP_Array_Initialization_Updates	NP-EMD.2006.510.0029 Rev. ---	2 Jun 2006
NGST/SE technical memo – NPP_VIIRS_CTP_FIX_WindowIR_Search	NP-EMD.2006.510.0093 Rev. ---	5 Dec 2006
NGST/SE technical memo – NPP_VIIRS_CTP_Exclude_Heavy_Aerosol_Pixels	NP-EMD.2008.510.0004 Rev. ---	17 Jan 2008
Data Processor Inter-subsystem Interface Control Document (DPIS ICD)	D35850 Rev. U.2	27 Aug 2008
D35836_G_NPOESS_Glossary	D35836_G Rev. G	10 Sep 2008
D35838_G_NPOESS_Acronyms	D35838_G Rev. G	10 Sep 2008

1.3.2 Source Code References

The science and operational code and associated documentation relevant to the algorithms described in this OAD are listed in Table 2.

Table 2. Source Code References

Reference Title	Reference Tag/Revision	Revision Date
Cloud Top Parameters Software Version Description	P1187-SW-I-008 Ver. 1.0	2 Sep 2003
Cloud Top Parameters Unit Test Plan and Report	P1187-Sw-I-009 Ver. 1.0	1 Sep 2004
VIIRS CTP IP Science Grade Software Unit Test Document	20080408_ISTN_VIIRS_NGST_3.5.3_DATA D39567 Rev. B	6 Mar 2008
VIIRS CTP operational software	B1.5.x.1	Jul 2008
NGST/SE technical memo – NPP_VIIRS_CTP_FIX_WindowIR_Search	NP-EMD.2006.510.0093 Rev. ---	5 Dec 2006
NGST/SE technical memo – NPP_VIIRS_CTP_Array_Initialization_Updates	NP-EMD.2006.510.0029 Rev. ---	2 Jun 2006
NGST/SE technical memo – NPP_VIIRS_CTP_Exclude_Heavy_Aerosol_Pixels	NP-EMD.2008.510.0004 Rev. ---	17 Jan 2008
NGST/SE technical memo – NPP_VIIRS_CTP_array_out_of_bound_fix	NP-EMD.2008.510.0048 Rev. ---	27 Aug 2008

2.0 OPERATIONAL ALGORITHM DESCRIPTION

The purpose of this Cloud Top Parameters (CTP) module is to estimate the Cloud Top Temperature (CTT), Cloud Top Pressure (CTp), and Cloud Top Height (CTH) IPs based on VIIRS radiances, other VIIRS cloud EDRs, and an ancillary profile of atmospheric temperature and moisture. The CTT from the Cloud Optical Properties (COP) module is used for ice clouds (day and night) and for water clouds at night. Once CTT is established, CTH is determined using linear interpolation based on the ancillary temperature sounding, while CTp is computed using the hypsometric equation. For daytime water clouds, CTp is determined by matching observed IR radiances to results from forward model calculations, while CTT and CTH are determined by interpolation. Figure 1 illustrates the algorithm processing chain, identifying all CTP inputs (including internal databases) and IP outputs.

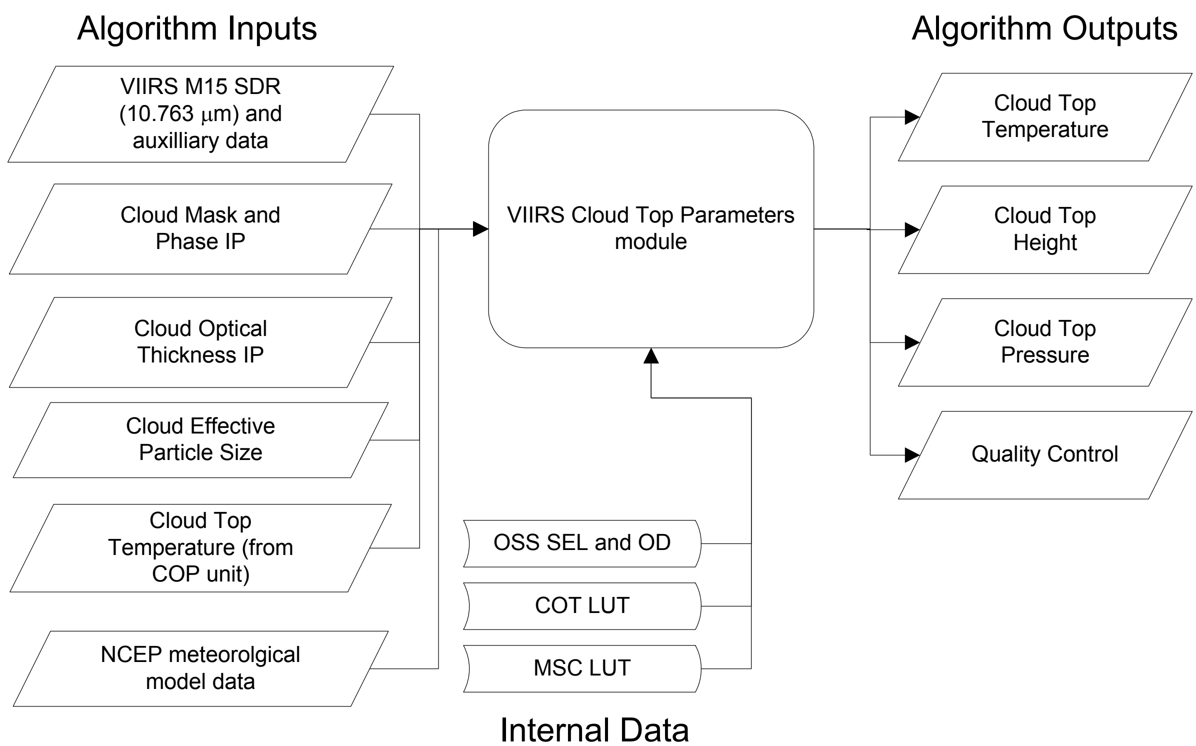

Figure 1. Cloud Top Parameters Processing Chain

2.1 Cloud Top Parameters (CTP) EDR Description

The CT retrieval algorithm and the theoretical basis are described in detail in the VIIRS Cloud Top Parameters Algorithm Theoretical Basis Document (ATBD), D43754. The CT module fully implements all features identified in the Cloud Top Parameters ATBD.

2.1.1 Interfaces

The CTP algorithm is designed to interface with inputs/outputs retrieved from the Data Management System (DMS) in binary format. The files contain moderate-resolution pixel-level data and products in the sensor projection. The size of the data files is specified in terms of the number of lines along-track and the number of pixels in the scan direction. Furthermore, data items from ING also include the specification of configurable threshold values.

2.1.1.1 Inputs

The inputs to the CTP module are summarized in Table 3 through Table 8. All input data is in binary format, retrieved from DMS. This includes

- Sensor Data Record (SDR) and auxiliary data (i.e., latitude, longitude, etc) obtained from the VIIRS SDR Module
- VIIRS Cloud Mask (VCM) Intermediate Product (IP) data obtained from the VCM Module
- Cloud Optical Thickness (COT) and Effective Particle Size (EPS) IP data obtained from the VIIRS COP Module
- CTT determined from University of California, Los Angeles (UCLA) algorithm obtained from the VIIRS COP Module
- Ancillary meteorological data interpolated to the VIIRS moderate-resolution pixel grid

In each table, the variable type is identified, a short description is provided, and the units and range for the variable are identified. No range checking of these inputs is performed in the code. The expected, valid range of inputs is provided for reference.

Table 3 summarizes the SDR and Auxiliary inputs to the algorithm. Only M15 radiances and brightness temperatures are required by the algorithm. The data is read as unscaled values.

Table 3. CTP SDR and Auxiliary Inputs

Input	Type	Description/Source	Units	Valid Range
Radiance_Mod	Float32 [M_VIIRS_SDR_ROWS][M_VIIRS_SDR_COLS]	VIIRS SDR radiance for channel M15	W/m^2-sr-μm	0-27
BrightTemp_Mod	Float32 [M_VIIRS_SDR_ROWS][M_VIIRS_SDR_COLS]	VIIRS SDR brightness temp for channel M15	K	100-390
SenZenAng_Mod	Float32 [M_VIIRS_SDR_ROWS][M_VIIRS_SDR_COLS]	Sensor zenith angle at each pixel from VIIRS SDR MOD geolocation structure	radians	0 to π
SolZenAng_Mod	Float32 [M_VIIRS_SDR_ROWS][M_VIIRS_SDR_COLS]	Solar zenith angle at each pixel from VIIRS SDR MOD geolocation structure	radians	0 to π

The CTP VCM inputs are summarized in Table 4. The information from the cloud mask is extracted in the doProcessing() method. The information extracted from the VCM includes Cloud Phase, Cloud Confidence, Cloud Mask Quality, Sun Glint Flag, Non Cloud Obstruction Flag, Snow or Ice Surface Flag, and Land/Water Background.

Table 4. CTP Cloud Mask Inputs

Input	Type	Description/Source	Units	Valid Range
VIIRSCloudMask	UInt8 [M_VIIRS_SDR_ROWS][M_VIIRS_SDR_COLS]	VIIRS Cloud Mask data for each pixel	Unitless	N/A

The COT inputs are summarized in Table 5. The COT, EPS, plus the quality flag for the products are read. The data is read as unscaled values.

Table 5. CTP Cloud Optical Thickness Inputs

Input	Type	Description/Source	Units	Valid Range
COT_IP	Float32 [M_VIIRS_SDR_ROWS][M_VIIRS_SDR_COLS]	Cloud optical thickness IP	Unitless	0.1 to 64 [Note A]

Input	Type	Description/Source	Units	Valid Range
EPS_IP	Float32 [M_VIIRS_SDR_ROWS][M_VIIRS_SDR_COLS]	Cloud effective particle size IP	microns	0-50 μm [Note A]
COP_IP_Quality Byte 0	UInt8 [M_VIIRS_SDR_ROWS][M_VIIRS_SDR_COLS]	Cloud optical properties quality control	Unitless	0 to 10
COP_IP_Quality Byte 1	UInt8 [M_VIIRS_SDR_ROWS][M_VIIRS_SDR_COLS]	Cloud optical properties quality control	Unitless	0 to 10
Note A – Data are only used for water phase clouds (if daytime). Range given is from NPOESS specification. Algorithm will function properly with any data within this range. Expected water phase cloud effective particle size is within range 3-15 μm the vast majority of the time.				

The CTT inputs as determined by the UCLA algorithms within the COP module are summarized in Table 6. The CTT and corresponding quality flag are required. The data is read as unscaled values.

Table 6. CTT Inputs

Input	Type	Description/Source	Units	Valid Range
INWCTT_IP	Float32 [M_VIIRS_SDR_ROWS][M_VIIRS_SDR_COLS]	UCLA algorithm determination of cloud top temperature	K	175 to 310
INWCTT_IP_Quality	UInt8 [M_VIIRS_SDR_ROWS][M_VIIRS_SDR_COLS]	Quality control for cloud top temperature determination	Unitless	0-31

The required ancillary inputs derived from meteorological data are summarized in Table 7 and Table 8. The atmospheric data is expected on the VIIRS moderate resolution sensor projection. The height profile and tropopause height inputs are converted from m to km for the use of this algorithm.

Surface pressure is determined by computing the geopotential height based on terrain height and latitude, then interpolating in log pressure to find the corresponding pressure at this height.

Table 7. Ancillary Inputs

Input	Type	Description/Source	Units	Valid Range
Height_Profile	Float32 [M_VIIRS_SDR_ROWS][M_ VIIRS_SDR_COLS][NCEP_ LAYER_ENUM_SIZE]	Height profile for specified pressure levels at all grid points	km (input m, convert m to km)	0-100
Temperature_Profile	Float32 [M_VIIRS_SDR_ROWS][M_ VIIRS_SDR_COLS][NCEP_ LAYER_ENUM_SIZE]	Temperature profile for specified pressure levels at all grid points	K	0-400
Moisture_Profile	Float32 [M_VIIRS_SDR_ROWS][M_ VIIRS_SDR_COLS][NCEP_ LAYER_ENUM_SIZE]	Moisture profile for specified pressure levels at all grid points	g/kg	0-20
Tropopause_Height	Float32 [M_VIIRS_SDR_ROWS][M_ VIIRS_SDR_COLS]	Tropopause height for all grid points	km (input m, convert m to km)	0-100
Sfc_Temp	Float32 [M_VIIRS_SDR_ROWS][M_ VIIRS_SDR_COLS]	Surface temperature for all grid points	K	0-400

Table 8. Additional Ancillary Inputs

Input	Type	Description/Source	Units	Valid Range
Surface_Pressure	Float32 [M_VIIRS_SDR_ROWS][M_VIIRS_SDR_COLS]	Surface pressure for all grid points	hPa (mbar)	0-1100
TerrainHgt_Mod	Int16 [M_VIIRS_SDR_ROWS][M_VIIRS_SDR_COLS]	Terrain height at each pixel	meters	1000-10000

2.1.1.1.1 Internal Data

A handful of look-up tables (LUTs) and algorithm databases are used by the CT day/water algorithm. Table 9 describes the key internal data used by the various algorithm components.

These tables are subject to revision based on scientific advances in the underlying algorithms or NPOESS testing and validation. They should be reviewed prior to launch and periodically during operations to determine if updated tables should be incorporated. In particular, the coefficients used in the Optimal Spectral Sampling (OSS) model are based on training that assumes a square response shape for the M15 band. Once the actual shape is determined, the OSS coefficients should be regenerated to reflect the new information.

Table 9. Summary of Key Internal Data (static) Used by the Day/Water Algorithm

Data	Description	Used in	Name
COT LUT	Contains factor for conversion of COT at 0.55 μm to 10.763 μm	ProcessDayWaterCloud()	COTlut.db
OSS SEL	OSS forward model input parameters	ComputeRTM()	sel645-1150-v.cmpr
OSS OD	OSS forward model input parameters	ComputeRTM()	od_885-971-v-24upd.cmpr
MSC LUT	Contains regression coefficients for multiple scattering correction	ComputeRTM()	RDATA_MSC_coef.dat

2.1.1.1.2 Requirements for Input

The retrieval of CTP follows execution of the VIIRS CloudMask/Phase algorithm and the UCLA COP algorithm. These algorithms provide the necessary inputs describing the cloud mask, cloud phase and CTT (for non daytime water clouds) for each pixel.

Auxiliary inputs are obtained from the VIIRS SDR processing. No special requirements are imposed on this data.

Ancillary inputs are derived from NCEP Numerical Weather Prediction (NWP) data. Nearest neighbor interpolation from the latest forecast model should suffice.

2.1.1.2 Outputs

The CTP module writes output products to a binary file and sends it to DMS. The output products include the CTT, CTH, and CTp products, plus quality control for each pixel. These are described in Tables 10 and 11. Other field attributes (identified in the Detail Design Document, DDD) describing the units, valid range, and fill values have not been included in the current algorithm.

Table 10. Output CTP IP Content

Output	Data Type/size	Description	Units	Valid Range
CTT_IP	Float32 [M_VIIRS_SDR_ROWS][M_VIIRS_SDR_COLS]	Cloud Top Temperature product	K	175 to 310
CTH_IP	Float32 [M_VIIRS_SDR_ROWS][M_VIIRS_SDR_COLS]	Cloud Top Height product	km	0 to 20
CTP_IP	Float32 [M_VIIRS_SDR_ROWS][M_VIIRS_SDR_COLS]	Cloud Top Pressure product	hPa (mbar)	50 to 1050
CTParm_IP_Quality Byte 0	UInt8 [M_VIIRS_SDR_ROWS][M_VIIRS_SDR_COLS]	Quality control for CTP products	Unitless	0 to 10
CTParm_IP_Quality Byte 1	UInt8 [M_VIIRS_SDR_ROWS][M_VIIRS_SDR_COLS]	Quality control for CTP products	Unitless	0 to 10
CTParm_IP_Quality Byte 2	UInt8 [M_VIIRS_SDR_ROWS][M_VIIRS_SDR_COLS]	Quality control for CTP products	Unitless	0 to 10

Table 11. Output CTP Quality Flag Specifications

Quality Flag – Byte 0	Bit Field	Description/Source	Value
qf_ctp_altitude	0-1	Altitude range check	1: < 3; 2: >3 && < 7; 3: > 7 (km)
qf_ctp_surfType	2-4	Surface type	0: land desert; 1: land not desert; 2: inland water; 3: sea water; 5: coastal
qf_ctp_sunglint	5	Sun glint	1: in sun glint; 0: not
qf_ctp_range	6	Check if out of range (50-1050 mb)	1: out of range; 0: not
qf_ctp_snowice	7	Snow/ice surface	1: snow/ice; 0: not
Quality Flag – Byte 1	**Bit Field**	**Description/Source**	**Value**
qf_ctp_cld_phase	0-2	Cloud phase 0 corresponds to not executed, 2 corresponds to cirrus and Opaque ice from VCM	0: not exec; 1: water; 2: ice; 3: mixed
qf_cth_range	3	Check if out of range (0-20 km)	1: out of range; 0: not
qf_ctt_range	4	Check if out of range (175-310 Kelvin)	1: out of range; 0: not
qf_residual_night_water	5	IR ice ctt convergence	1: convergence; 0: not
qf_residual_night_ice	6	IR ice ctt convergence	1: convergence; 0: not
qf_residual_ir_day_ice	7	IR ice ctt convergence	1: convergence; 0: not
Quality Flag – Byte 2	**Bit Field**	**Description/Source**	**Value**
qf_ctp_runbackup	0-2	A quality flag for using black cloud approach	0: clear; 1: non-day/water convergence; 2: day/water convergence; 3: black cloud 1 convergence; 4: black cloud 2 in bound; 5: black cloud 3 (not used); 7: nonconvergence

2.1.1.2.1 Requirements for Output

The CTP IPs described here are used as inputs to other cloud module software units. Specifically, the CTP IP outputs are input directly to the Perform Parallax Correction (PPC) software unit and then indirectly to the Cloud Cover/ Layers (CCL), Cloud Base Height (CBH) and the Grid Cloud EDR software units.

The cloud CTH IP product is defined as geopotential height. See Appendix A. Definition of Cloud Top

2.1.2 Algorithm Processing

The objective of the CTP algorithm is to determine CTT, CTp, and CTH for all cloudy pixels in a dataset. A conceptual view of the processing of the CTP IP indicating processing in both the COP Unit and CTP Unit is given in Figure 2 presents the dataflow model matching the code. The names in the bubbles are the actual function names in the code. These are described later. In Figure 2 the functions from Begin through InitRTM() read in the input data and initialize the program.

The module DetermineProcessPath() identifies pixels for processing by identifying them as cloudy and free of sun glint and categorizes them as either daytime water clouds, or not daytime water clouds. If the cloud is identified as “day/water” then the retrieval of CTT, CTp, and CTH is carried out using the ProcessDayWaterCloud() routine. The ProcessNonDayWaterCloud() module determines the CTP for all other cloudy pixels.

The primary scientific functions are contained in the ProcessDayWaterCloud and ProcessNonDayWaterCloud functions.

ProcessNonDayWaterCloud is the simpler of the two. It performs a hydrostatically consistent interpolation to derive the CTH and CTp using CTT (derived previously by the COP Unit as a primary input).

The Window IR Daytime Water retrieval algorithm conceptual flow is illustrated in Figure 4. It uses a physical retrieval algorithm with embedded RT model to derive the CTp. Then interpolation methods are used to compute the CTT and CTH.

The quality of the retrievals is assessed in the module ComputeParmQuality(). Finally, the results are written to the DMS and the program terminates cleanly.

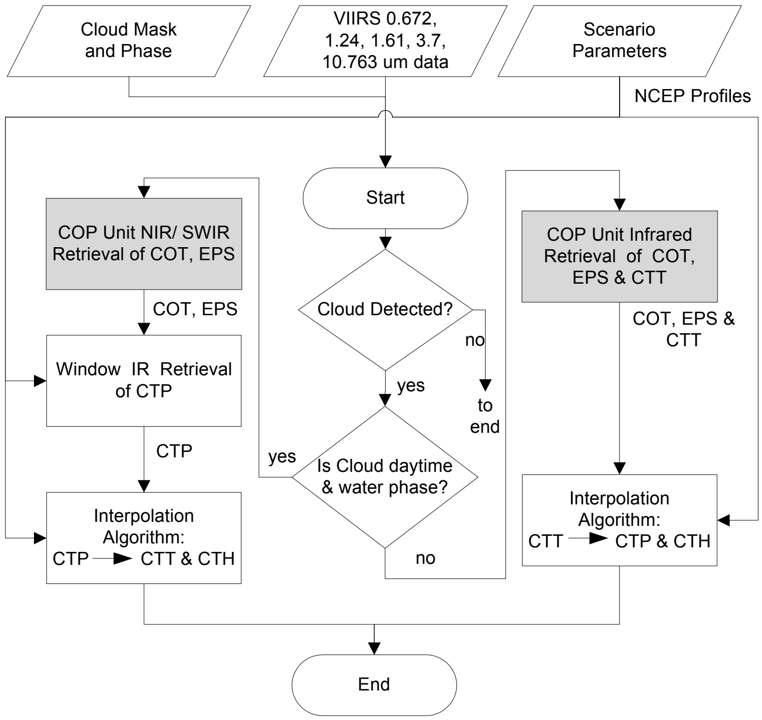

Figure 2. Conceptual Process for Cloud Top Parameter Retrieval

(shaded cells indicated steps performed in the COP Algorithm)

The following subsections describe the functions of Figure 3. Calling arguments are described and a brief description of functionality is given. Additional details are contained in the code header files and the CTP Unit DDD.

Where the text refers to a return code per standard code convention, it refers to a return of either PRO_SUCCESS (if no error encountered) or PRO_FAIL (if errors encountered). The values are constants defined in ProCmnDefs.h.

Unless otherwise stated the functions are coded in C/C++. There are also many FORTRAN90 functions/ subroutines.

The following function descriptions adopt the convention that if a value is referenced inherently as a pointer, then the variable name is given as *var_name and the type given as type. If a pointer is used as an array or string, then the variable name is given as var_name and the type is given as type*.

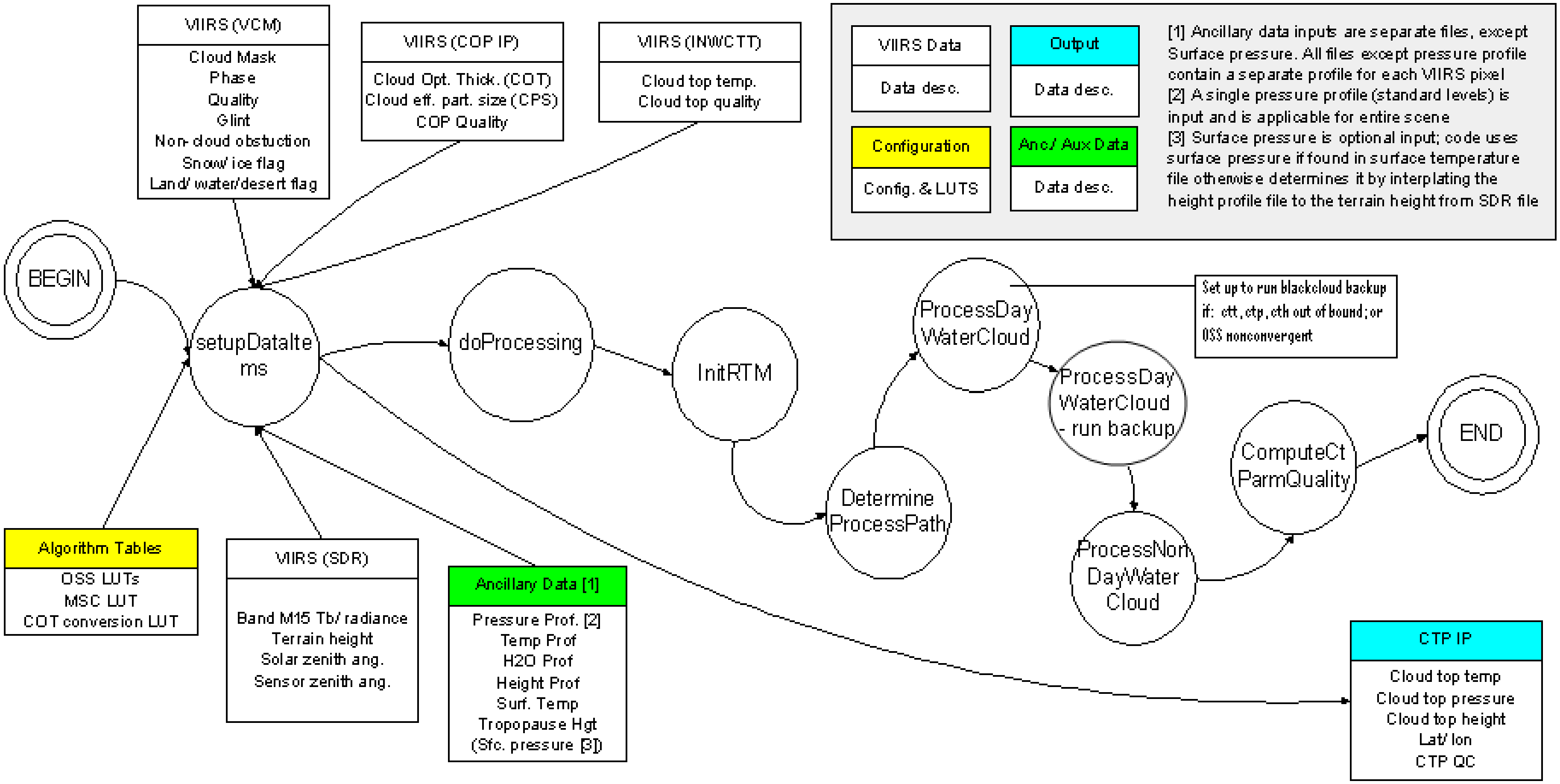

Figure 3. Dataflow Model of the CTP IP Software

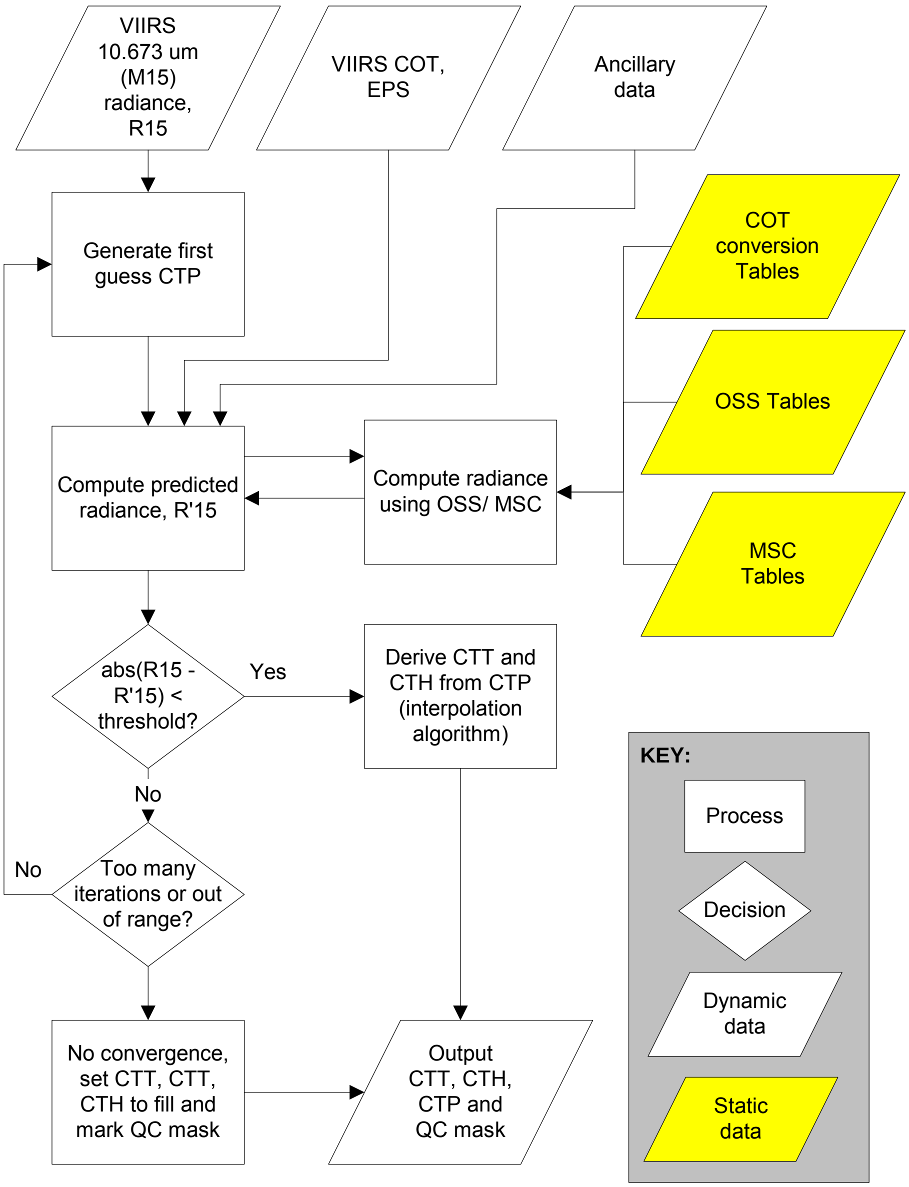

Figure 4. Window IR Daytime Water Cloud CTT Retrieval Algorithm Logical Flow

2.1.2.1 Main Module – doProcessing()

This is the overridden base class function that calls the actual CTP algorithm code.

2.1.2.2 InitRTM()

Return Type: Int32

Table 12 describes the function argument variables for InitRTM.

Table 12. InitRTM Function Argument Variables

Name	Type	Initialization	Description
*Rtm	RTM_STRUCT	With data read in from OSS LUTs	Pointer to structure containing data input from OSS LUTs and COT conversion LUT
*ctpDataPtr	ViirsCtpDataType	NA	Pointer to structure containing pointers to input and output structures

Data read in from the COT wavelength correction LUT is stored in the Rtm structure. Data read in from the OSS LUTs is stored in the Rtm structure.

Return code per standard convention.

Sub-functions: OssIn_Ir
See detail below.

2.1.2.2.1 OssIn_Ir()

This is a FORTRAN90 subroutine.

Return Type: <none>

Table 13 describes the input argument variables for OssIn_Ir.

Table 13. OssIn_Ir Input Argument Variables (given in C conventions)

Name	Type	Initialization	Description
*ossOdPtr	ViirsCtpOssOdType	none	Pointer to structure containing data from the OSS OD LUT
*ossSelPtr	ViirsCtpOssSelType	none	Pointer to structure containing data from the OSS SEL LUT
nlev_in	Int32	none	Number of levels for assumed by remainder of program
nchan_in	Int32	none	Number of channels assumed by remainder of program
nnodes_out	Int32	set from OSS LUTs	Total number of “nodes” in the OSS model
nchan_out	Int32	set from OSS LUTs	Actual number of channels in the OSS model
nlev_out	Int32	set from OSS LUTs	Actual number of levels in the OSS model
pref_out	Float32 *	set from OSS LUTs	Pressure (in mb), 1-D array of size nlev_out used
frq_out	Float32 *	set from OSS LUTs	Frequencies (in cm-1) for each node of the OSS LUT, 1-D array of size nnodes_out used
iret	Int32	set based on return status	Return status: 1 = OK 3 = different number of levels found in LUT than assumed 4 = different number of channels found than assumed

This function returns values from the OSS LUTs so they can be stored in the RTM_STRUCT for use by the working OSS calls.

It is currently required that the assumed input size for number of levels and number of channels be the same as that found in the referenced OSS LUTs. The calling of this subroutine currently has these as MXLEV and MXCHAN, respectively (as defined in ctp.h).

Return status is contained in the calling arguments. See Table 13.

2.1.2.3 DetermineProcessPath()

Return Type: int

Table 14 describes the input argument variables for DetermineProcessPath.

Table 14. DetermineProcessPath Input Argument Variables

Name	Type	Description
*config	IngMsdCoefficients_ViirsCtpStruct	Configurable Parameters (i)
*Mask	CM_STRUCT	Cloud mask (i)
*Sdr	SDR_STRUCT	SDR data (i)
*Ips	IP_STRUCT	IP data from COP Unit (i)
*Work	WORK_STRUCT	Work structure (io)

Return code per standard convention.

This function sets a flag in the work array indicating which processing path is appropriate for each pixel. Pixels marked as “day/water” (indicated by the presences of the CTT produced by the COP Unit) are tagged for processing by the ProcessDayWaterCloud() function. Other cloudy pixels (confident cloudy indicator used) are marked for processing by ProcessNonDayWaterCloud(). Other pixels are inserted with fill values.

2.1.2.4 ProcessDayWaterCloud()

Return Type: int

Table 15 describes the input argument variables for ProcessDayWaterCloud.

Table 15. ProcessDayWaterCloud Input Argument Variables

Name	Type	Description
*config	IngMsdCoefficients_ViirsCtpStruct	Configurable Parameters (i)
*Mask	CM_STRUCT	Cloud mask (i)
*Sdr	SDR_STRUCT	SDR data (i)
*Aux	AUX_STRUCT	Auxiliary data (i)
*Ips	IP_STRUCT	IP data in from COP Unit (i)
*Work	WORK_STRUCT	Work structure (io)
*Rtm	RTM_STRUCT	OSS model data structure (i)
*mlut	MLUT_STRUCT	Multiple scattering correction structure (i)
*dwS	DWBLOCK_STRUCT	Work array containing variables/ space used specifically for the day/water processing path
BCflag	Int32	Black cloud flag

This function performs the basic science portion of the algorithm for daytime conditions and water phase clouds. It employs the Infrared Window Algorithm to compute the CTp directly using an iterative retrieval algorithm with embedded Radiative Transfer Model (RTM) (referred to also as a forward model). It then determines the associated CTT and CTH using interpolation methods applied to the NWP data in the *Aux structure.

Pixels identified with water clouds during the day and pixels with process paths matching “BCflag” (if the “BLACK_CLOUD_1” switch is defined, pixels marked as “RUN_BACKUP” will be processed) get processed by the ProcessDayWaterCloud() module. This is illustrated in Figure

5. This routine determines the CTP by comparing the observed TOA radiance in band M15 with calculations based on a forward model. The data buffer is processed in blocks. The size of the block is specified by the numAggAt and numAggXt parameters defined in ctp.h. For VIIRS data, the block size will be 8 by 8. Within a block, all pixels identified as “day/water” and containing valid SDRs, COT, EPS, etc are sorted by brightness temperature in increasing order. The purpose here is to use the brightness temperature with the lowest value to derive the first guess at the CTP for that pixel. Then, provided convergence is achieved, each successive solution to CTP within the block is used as the first guess for the next pixel in the list. For simulation data used in performance testing, the block size can be set to 1 x 1 so that the first guess is initialized for each pixel.

The forward model calculations are performed using the OSS method. Prior to the retrievals, meteorological model data used in the calculations is interpolated to the pressure grid used by the OSS model. Retrievals are performed for each pixel in the block that is identified with “day/water”. The retrieval approach is based on the Newton-Raphson method. First all the input data fields are loaded. Then the visible COT is converted to a corresponding infrared value based on a LUT and the surface emissivity is assigned. Then the RTM function ComputeRTM() is called which computes the top-of-atmosphere (TOA) radiance and radiance derivative (Jacobian) with respect to the CTp. The convergence is determined by comparing the difference in model and observed radiance to the noise amplitude for the M15 band. If the solution has not converged and if the “WINDOW_IR_SEARCH” switch has been defined, then CTp is updated using the radiance derivative determined from the forward model calculations and the iterations continue. The solution is constrained between the pressure at the surface and the pressure at the tropopause. If the solution has converged, then the derived value of CTp is saved and used to determine CTT and CTH. In this case, temperature and height are determined by interpolating the atmospheric profiles with respect to the log of pressure. If no solution is found after a set number of iterations, the processing path is set to “RUN_BACKUP”. This will trigger the retrieval of CTP using the blackcloud assumption in a later function call. Range checking is applied to the final retrieved values.

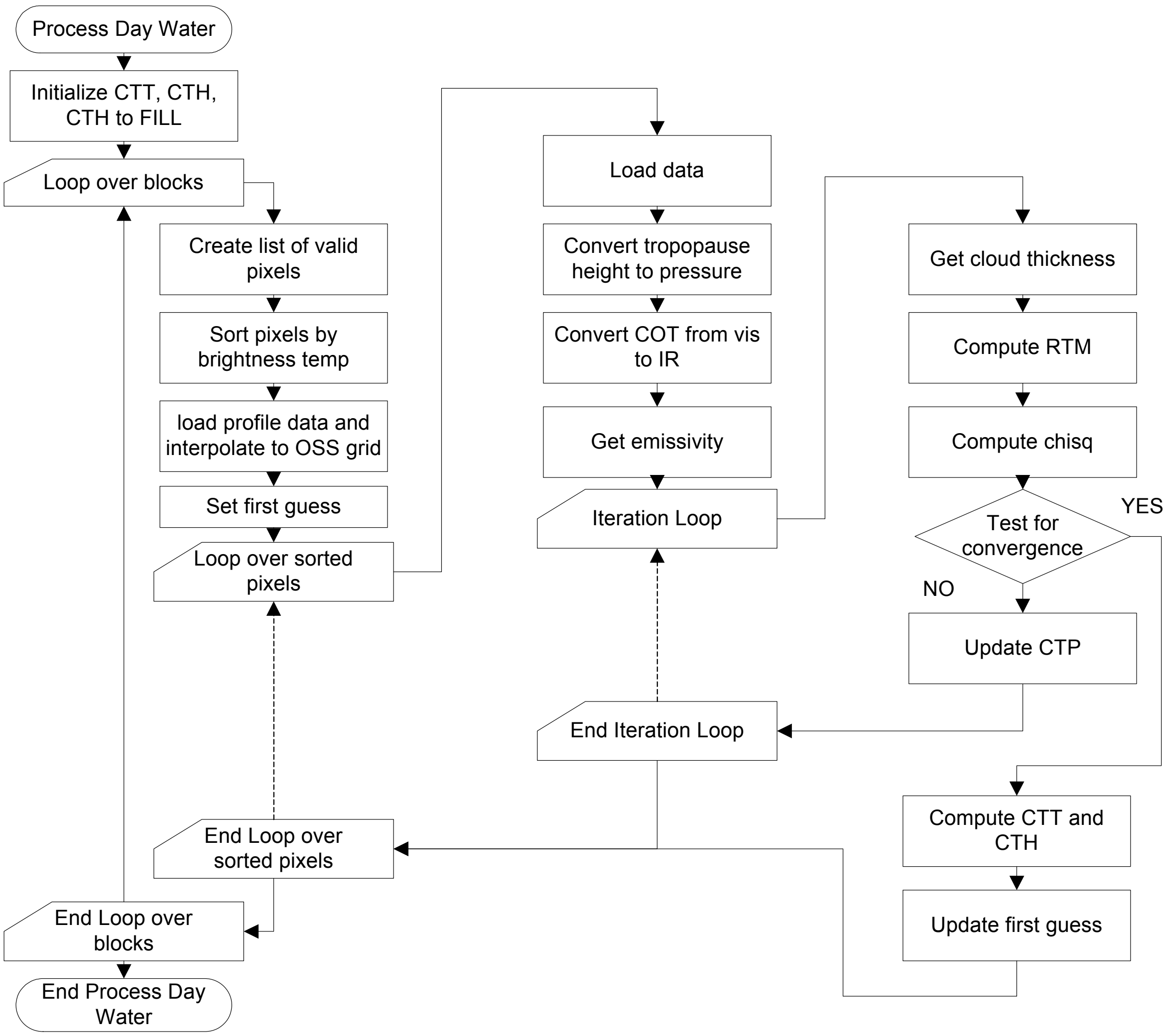

Figure 5. Process Day/Water Cloud Flow Diagram

Sub-functions: q_sort, interpTZ, interpToLevels, getFirstCTP, cotconvert, getEmissivity, computeRTM, getthick

Details of these functions follow.

2.1.2.4.1 q_sort()

Return Type: <none>

Table 16 describes the input argument variables for q_sort.

Table 16. q_sort Input Argument Variables

Name	Type	Description
data	Float32*	Values to be sorted from smallest to largest (io)
index	Int32*	Array of size N, of the indexes pointing to the positions in the work array of the values to be sorted (io)
N	Int32	Size of array to be sorted (i)

This recursive function sorts the values in the array data from smallest to largest using a quick-sort algorithm. The variables in the array index, which point to the location in the work arrays where each value is contained, are moved at the same time. On output the index array contains the indexes sorted from smallest to largest.

The function is used to sort the pixels in a local region by increasing brightness temperature of band M15. The day/water algorithm uses this to pick a first guess and thus minimize number of iterations.

2.1.2.4.2 interpTZ()

Return Type: Int32

Table 17 describes the input argument variables for interpTZ.

Table 17. interpTZ Input Argument Variables

Name	Type	Description
inP	Float32	Cloud top pressure (i)
p	Float32*	Array of pressures in mb (i)
t	Float32*	Array of temperatures (i)
z	Float32*	Array of heights (i)
pSfc	Float32	Surface pressure (mb) (i)
tSfc	Float32	Surface temperature in K (i)
zSfc	Float32	Surface height (i)
T	Float32*	Cloud top temperature (o)
Z	Float32*	Cloud top height(o)
N	Int32	Number of pressure levels

This function interpolates the CTp value derived using the Infrared Window Algorithm to determine the CTT and CTH values.

The function always returns PRO_SUCCESS.

2.1.2.4.3 interpToLevels()

Return Type: Int32

Table 18 describes the input argument variables for interpToLevels.

Table 18. interpToLevels Input Argument Variables

Name	Type	Description
config	IngMsdCoefficients_ViirsCtpStruct*	Configurable Parameters (i)
inP	Float32*	Array of size inN containing pressure in mb (i)
inT	Float32*	Array of size inN containing temperature at inP levels in K (i)
inW	Float32*	Array of size inN containing water vapor at the inP levels in g/kg (i)
pSfc	Float32	Surface pressure (i)

Name	Type	Description
outP	Float32*	Array of size outN containing pressure in mb (i)
outT	Float32*	Array of size outN containing temperature at the outP levels in K (o)
outW	Float32*	Array of size outN containing water vapor at the inP levels in g/kg (o)
inN	Int32	Number opf vertical levels in the input arrays (i)
outN	Int32	Number of vertical levels in the output arrays (i)

This function interpolates data from the NWP vertical grid to the levels used by the OSS model.

The function always returns PRO_SUCCESS.

2.1.2.4.4 getFirstCTP()

Return Type: Float32

Table 19 describes the input argument variables for getFirstCTP.

Table 19. getFirstCTP Input Argument Variables

Name	Type	Description
t	Float32	Input brightness temperature
P	Float32*	Pressure profile in mb, size n
T	Float32*	Temperature profile in K, size n
n	Int32	Number of levels in profiles

The function generates a basic first guess CTp for each processing region. It determines at which pressure level the brightness temperature most closely matches the atmospheric temperature. This basic first guess is only used for the first time in each processing region.

The return value is the first guess CTp.

2.1.2.4.5 COTconvert()

This is a FORTRAN90 subroutine. Table 20 describes the input argument variables for COTconvert.

Return Type: <none>

Table 20. COTconvert Input Argument Variables (C conventions)

Name	Type	Description
nCOT	Int32	Number of actual COT look-up table entries (i)
COTlut	Float32*	The COT LUT, a2-D array of size [mxcot][2], where the first pair of entries is the EPS (μm) and the second is the conversion factor from COT from the visible reference produced by the COT IP Unit and the output is the infrared value at the band M15 wavelength (i)
CPS	Float32	The cloud particle size in μm (i)
COT	Float32	The input (visible) cloud optical thickness, unitless (i)
COTir	Float32	Infrared COT from above conversion (o)

The size of the COTlut array dimension mxcot is defined in oss.h.

2.1.2.4.6 getEmissivity()

Return Type: Float32

Table 21 describes the input argument variables for getEmissivity.

Table 21. getEmissivity Input Argument Variables

Name	Type	Description
cmSNowIce	uint8	VCM for snowice or no snowice (see vcm_flags.h) (i)
cmSfcType	unint8	VCM for surface type (see vcm_flags.h) (i)

This function is passed the VIIRS Cloud Mask values for snow/ice and a surface type and assigns a surface emissivity value (see ctp.h). The return value is the surface emissivity.

2.1.2.4.7 getthick()

Return Type: Float32

Table 22 describes the input argument variables for getthick.

Table 22. getthick Input Argument Variables

Name	Type	Description
*config	IngMsdCoefficients_ViirsCtpStruct	Configurable Parameters (i)
cot	Float32	Cloud optical thickness IP from COP Unit (i)
ctp	Float32	Cloud to pressure guess (i)

This function returns an estimate of the cloud thickness (in mb) given the COT from the COP Unit and the current guess of the CTp. The thickness is the return value.

2.1.2.4.8 computeRTM()

Return Type: Int32

Table 23 describes the input argument variables for computeRTM.

Table 23. computeRTM Input Argument Variables

Name	Type	Description
*mlut	MLUT_STRUCT	Multiple scattering LUT structure (i)
Tprof	Float32*	Temperature profile, K, at nlev levels (i)
Wprof	Float32*	Water vapor profile, g/kg, at nlev levels (i)
Psfc	Float32	Pressure, mb, at nlev levels (i)
Tsfc	Float32	Surface temperature, K (i)
Esfc	Float32	Surface emissivity (i)
cbp	Float32	Cloud base pressure, mb, estimate (i)
cot	Float32	Cloud optical thickness in the visible (i), but not used
cotir	Float32	Cloud topical thickness in the IR, band M15 wavelength
nlev	Int32	Number of levels in the input profiles
nchan	Int32	Number of channels computed by the OSS model (i)
frq	Float32*	Wavelengths of the channels computed by the OSS model
Lmod	Float32*	Radiance computed by the OSS model (mw/m^2/str/cm^{-1}) (o)
dLdPmod	Float32*	Derivative of radiance with respect to CTP computed by the OSS model (mw/m^2/str/cm$^{-1/mb}$) (o)
ctt	Float32*	Cloud top temperature (i)
MSCflag	Int32	Multiple scattering correction flag

Always returns PRO_SUCCESS.

This function is the driver routine for the OSS fast RTM. The next subsection provides an overview of the model.

2.1.2.4.8.1 OSS Fast Forward Model

The RTM used by the day/water algorithm consists of two components. The core of the algorithm is OSS, a fast forward model that addresses the radiative effects from the atmosphere, surface, and cloud. The effects of multiple scattering within a cloud are addressed by a regression correction to the OSS radiances. One of the inputs required by the model is the COT in the IR. A separate routine is provided that uses a LUT to convert COT at 0.55 μm to the M15 wavelength.

The OSS algorithm has been developed under the Fortran90 programming language. The RTM is initialized once up front prior to the execution of the CT retrievals. A single driver InitRTM() is introduced for this purpose. The routine calls OssIn_Ir() to initialize the OSS routines. These RTM initialization routines and their corresponding subroutines are described in Table 24.

Execution of the RTM calculations is carried out with a call to ComputeRTM(). This routine calls the OSS model, returning the TOA radiance for the given geophysical state plus the radiance derivatives wrt to the cloud pressure. Also produced are the radiance contributions directed into the cloud from below, into the cloud from above, and out of the cloud from above. These parameters are used by the MS regression routine to produce the final modeled TOA radiance. The main drivers within this routine are

- CompOssRad_Ir() which carries out the OSS calculations, and
- MScorrect() which applies the multiple scattering correction.

These RTM driver description and calling tree routines and their corresponding subroutines are described in Table 25

Table 25. The call and input/output list for compossrad_ir() and Mscorrect() are provided in Table 26Table 26 and Table 27Table 27. A flow diagram describing the RTM calculations is presented in Figure 6.

Table 24. RTM Initialization Description and Calling Tree

Calling Tree	Description
InitRTM()	Initialization for radiative transfer model calculations • performs initializations for COT conversion, OSS model, and MS correction
CALL OssIn_Ir()	Initialization wrapper for OSS forward model • calls OSS initialization routine
CALL oss_init_ir()	Initialization routine for the OSS forward model • returns geophysical vector points, frequencies, pressure grid etc.
CALL GetOD()	Reads the pre-computed Look Up Tables needed by the OSS radiative transfer model • open OD and SEL file and read headers • builds mapping vector for variable molecules • reads OSS parameters (from SEL) • reads absorption coefficient tables (from OD)
CALL init_kfix()	Initialization for fixed gases
CALL odthresh()	Compare optical depth of individual dry constituents to total optical depth and flag weakly absorbing molecules
CALL cum_fix()	Merges selected variable gases with fixed gases
CALL shrink_var()	Shrinks indices to reflect number of fixed gases

Table 25. RTM Driver Description and Calling Tree

Calling Tree	Description
ComputeRTM()	Driver for forward model calculations • Calls oss and msc and converts radiance to (W/m^2/μm/sr)
CALL CompOssRad_Ir()	Wrapper for OSS driver
CALL ossdrv_ir()	Driver for OSS radiative transfer model. Computes both radiances and their Jacobians wrt geophysical parameters. • Initializes radiance vector and k-matrix • Computes path variables o path geometry o avg temperature and molecular amounts for layers o coefficients for temperature interpolation of the ODs • Loops over spectral points o computes molecular optical depth for all layers o interpolates surface emissivity to node wavenumber o performs RT calculations o computes final radiance weighted over channels
CALL setpath_ir()	Compute items related to path geometry • Computes surface level • Sets viewing geometry parameters
CALL fpath_ir()	Calculate the average temperature and molecular amounts for all layers for given profiles of temperature and molecular concentrations. It also calculates the derivatives of tavl with respect to a change in the lower and upper boundary temperatures and the derivatives of the molecular amounts with respect to a change in the mixing ratios at the layer boundaries. Molecular amounts are in molec./cm**2. Integration assumes that T is linear in z (LnT linear in LnP) and LnQ linear in LnP. • computes average temperature for layers • calculates amounts for individual species and derivatives wrt mixing ratios for retrieved constituents
CALL lpsum()	Computes average layer quantities (or integrated amount) using a log-x dependence on log-p
CALL lint()	Interpolation in Log pressure
CALL settabindx()	Computes temperature/water vapor indexes and OD interpolation coefficients at each level
CALL osstc_ir()	Compute radiance derivative wrt to cloud top layer
CALL plank_set()	Calls Planck function for multiple layers
CALL planck()	Calculates Planck function and its derivative wrt temperature
CALL osstran()	Computes the layer optical depths at each level (owing to fixed gases, water, and variable gases) for a given LUT
CALL vinterp()	Interpolation in wavenumber
CALL ossrad()	Computes radiances (in mw/m2/str/cm-1) and derivatives of radiances with respect to atmospheric and surface parameters • computes Planck radiance at cloud top/bottom • computes Planck radiance for each layer • computes transmittance profile along viewing path • computes downwelling thermal radiance • adds contribution from the surface • computes upwelling thermal radiance • computes cloud top derivatives
CALL planck_int()	Accelerates Planck calculation by interpolation over frequencies intervals
CALL planck_set()	Computes Planck function for surface and average layer temperatures
CALL planck()	Calculates Planck function and its derivative wrt temperature
CALL MScorrect()	Apply multiple scattering correction • Determines regression coefficients by interp in LUT • Apply regression equation return corrected radiance
CALL MlutINterp()	Interpolation in LUT

Calling Tree	Description
CALL MlutValue()	Return a vector 'value' from a look-up table. This routine takes a vector index, and converts it into a linear index into the internal 'data' array. The output vector is copied into the passed 'value' array.

Table 26. Argument List for OSS Driver

Call	CompOssRad_Ir (Tprof, Wprof, nlev, Tsfc, Psfc, Esfc, vza, ctp, cbp, cotir, frq, chanWght, nchan, Lmod, dLdPmod, &taucldtop, &ctt_oss, &Ltop, &Ldwn, &Lbot)	
Inputs	Tprof = temperature profile (K)	Float32 *
	Wprof = water vapor profile ()	Float32 *
	nlev = number of levels of atmospheric data	Int32
	Tsfc = surface temperature (K)	Float32
	Psfc = surface pressure (mbar)	Float32
	Esfc = surface emissivity	Float32
	vza = view zenith angle (radians)	Float32
	ctp = clout top pressure (mbar)	Float32
	cbp = cloud base pressure (mbar)	Float32
	cotir = cloud optical thickness at M15 wavelength	Float32
	frq = frequencies	Float32 *
	chanWght = channel weights	Float32[nchan]
	nchan = number of channels	Int32
Outputs	Lmod = top-of-atmosphere radiance (mw/m^2/str/cm^{-1})	Float32
	DldPmod = radiance derivative wrt to cloud top (mw/m^2/str/cm^{-1}/mbar)	Float32
	Taucldtop = atmospheric opacity at top of cloud	
	Ctt_oss = cloud top temperature (K)	Float32
	Ltop = upward_directed radiance at cloud top (mw/m^2/str/cm^{-1})	Float32
	Ldwn = downward-directed radiance at cloud top (mw/m^2/str/cm^{-1})	Float32
	Lbot = upward-directed radiance at cloud bottom (mw/m^2/str/cm^{-1})	
		Float32
		Float32

Table 27. Argument List for MSC Driver

Call	Lms = MScorrect (mlut, vza, cot, cps, Ltop, Lbot, Ldwn);	
Inputs	mlut = includes all information about the MS look-up table (see mlut.h)	struct
	vza = view zenith angle (degrees)	
	cot = cloud optical thickness (at 0.55 mm)	Float32
	cps = cloud particle size (□m)	Float32
	Ltop = upward-directed radiance at cloud top (mw/m^2/str/cm^{-1})	Float32
	Lbot = upward-directed radiance at cloud bottom (mw/m^2/str/cm^{-1})	Float32
	Ldwn = downward-directed radiance at cloud top (mw/m^2/str/cm^{-1})	Float32
		Float32
Outputs	Lms = additive correction to radiance at the cloud top (mw/m^2/str/cm^{-1})	Float32

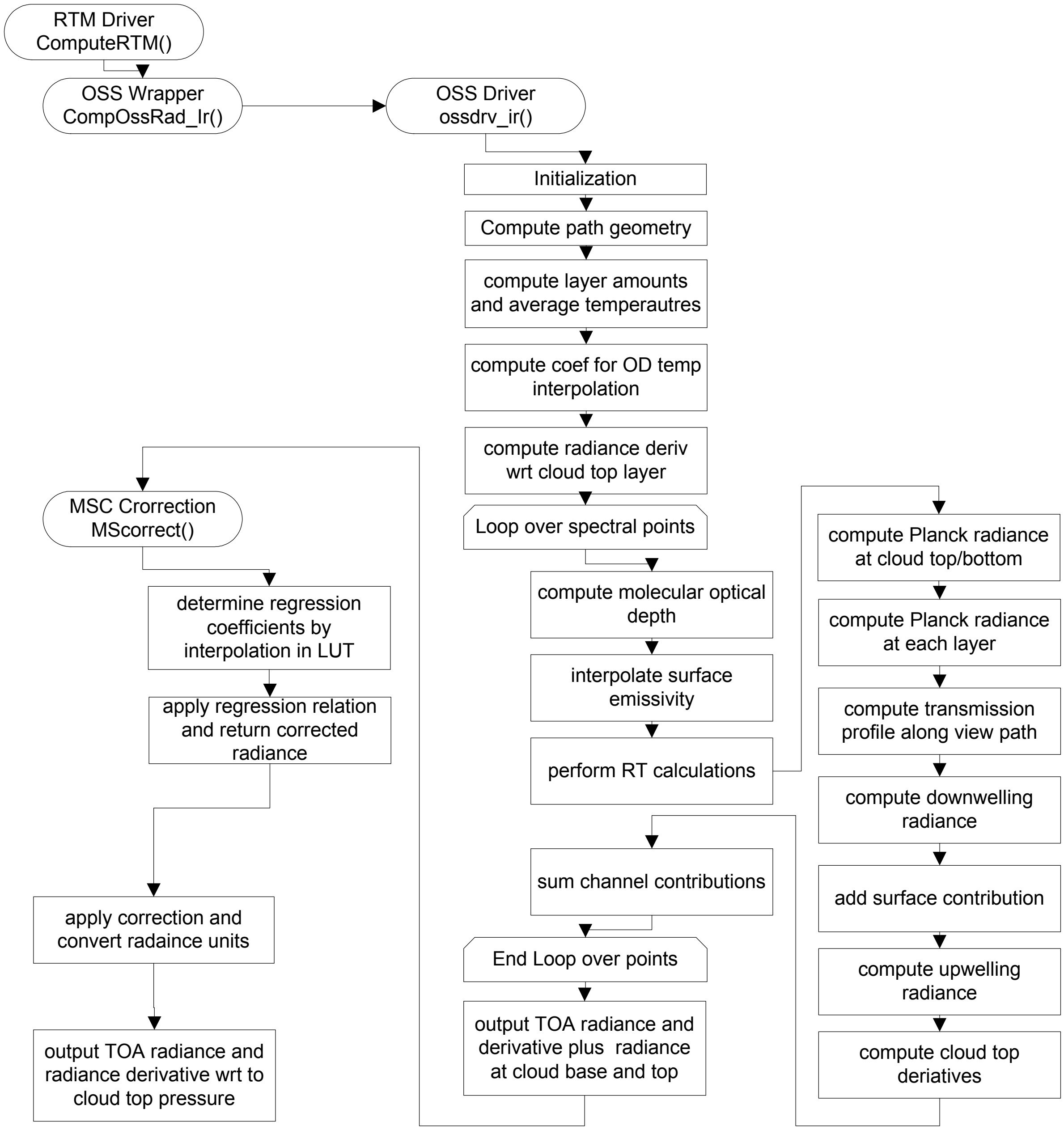

Figure 6. Flow Diagram for RTM Calculations

2.1.2.5 ProcessNonDayWaterCloud()

Return Type: Int32

Table 28 describes the input argument variables for ProcessNonDayWaterCloud.

Table 28. ProcessNonDayWaterCloud Input Argument Variables

Name	Type	Description
*Aux	AUX_STRUCT	Aux (weather forecast model) data structure (i)
*Sdr	SDR_STRUCT	SDR data (i)
*Ips	IP_STRUCT	IP structure containing COP Unit data (i)
*work	WORK_STRUCT	Work structure (io)

Return code per standard convention.

Non-Day/water retrievals are based on the CTT as determined by the UCLA COT retrieval algorithm. The determination of CTH and CTp is performed in the ProcessNonDayWaterCloud() module. This routine loops over all pixels in a buffer. If a given pixel has a valid CTT then processing begins. First the value of CTT is compared against the minimum and maximum atmospheric temperatures below the tropopause. If the CTT falls outside this range by more than 5 K then no retrievals of CTp or CTH are performed. If CTT lies within 5 K of the maximum then CTH and CTp are assigned values corresponding to the maximum temperature. If CTT lies within 5 K of the minimum temperature, then CTp and CTH are assigned values corresponding to the minimum temperature. If the CTT lies between the minimum and maximum temperature, then the algorithm attempts to find the best of all possible solutions for CTp and CTH. For each level of atmospheric data, if the CTT falls between the minimum and maximum temperatures from consecutive levels then the height and the cloud top dew point are determined by linear interpolation between levels. The dew point depression is computed as the difference between the dew point temperature and the CTT. The CTp is computed using the hypsometric equation. Solutions with sufficient moisture for clouds are identified (i.e., with dew point depression less than 3 K). If no solutions meet these criteria, then the CTp is set equal to the minimum and CTH is set to the maximum of all solutions. If multiple solutions meet this criteria, the most likely solution is identified as the one with maximum CTH and with minimum CTp.

2.1.2.6 ComputeCtParmQuality()

Return Type: Int32

Table 29 describes the input argument variables for ComputeCtParmQuality.

Table 29. ComputeCtParmQuality Input Argument Variables

Name	Type	Description
*config	IngMsdCoefficients_ViirsCtpStruct	Configurable Parameters (i)
*Mask	CM_STRUCT	Cloud mask structure (i)
*Work	WORK_STRUCT	Work structure (io)
*Ips	IP_STRUCT	IP structure containing COP Unit data (i)

Always returns PRO_SUCCESS.

This function computes the quality indicator. It sets a value from 1 to 10 (increasing confidence) based on the presence of sun glint, non cloud obstruction and the CM confidence indicator from the VCM.

2.1.2.7 Cloud Top Parameters Quality Flag Logic

The evaluation of CT quality is carried out in ComputeCtParmQuality(). The CTP quality flag is evaluated based on the presence of non-cloud obstruction and the values of the cloud mask confidence and quality. An additional test based on presence of sun glint (described in the CTP DDD) has not been implemented. The quality flag range is a scale from 0 to 10. FILL values are entered for pixels that are identified as confidently clear in the cloud mask. The default quality is 10. This is reduced by a factor of 0.8 if the cloud mask indicates a non-cloud obstruction. If the cloud mask identifies the pixel as probably clear, the quality is reduced by a factor of 0.3 times the cloud mask quality. If the pixel is identified as probably cloudy, the CTP quality is reduced by a factor of 0.7 times the cloud mask quality. Finally, if the pixel is identified as confident cloudy, the CTP quality is reduced by the cloud mask quality only. The quality reduction factors are specified in ctp.h. The ATBD identifies a number of other parameters that can influence the quality of the CTP retrievals. None of these have been incorporated in the current algorithm and should be considered in future versions.

2.1.2.8 Error Handling Logic

The current algorithm has limited error-handling logic. The ProcessDayWaterCloud() module performs limited range checking and tests for FILL on input data prior to processing. Valid "day/water" pixels are defined as those with valid M15 brightness temperatures, COT greater than zero, and temperature profiles where all values are above 100 K. Fill values are specified by parameters in the code and not determined from the input files themselves. The CTP retrievals are constrained to solutions below the tropopause height. The output quality flag reflects the quality of the cloud mask inputs only and does not reflect failures of the algorithm due to missing inputs.

2.1.3 Graceful Degradation

2.1.3.1 Graceful Degradation Inputs

There is one case where input graceful degradation is indicated in the CTP:

1. An input that is retrieved for an algorithm has the N_Graceful_Degradation metadata field set (propagation)

Table 30 details the instances of these cases. Note that the shaded cells indicate that the graceful degradation was done upstream at product production.

Table 30. Graceful Degradation

Input Data Description	Baseline Data Source	Primary Backup Data Source	Secondary Backup Data Source	Tertiary Backup Data Source	Graceful Degradation Done Upstream
Atmospheric Temperature Profile	VIIRS_GD_11.4.3 NCEP	VIIRS_GD_11.4.3 NCEP (Extended Forecast)	N/A	N/A	Yes
Corresponding Pressure Levels	VIIRS_GD_11.4.1 NCEP	VIIRS_GD_11.4.1 NCEP (Extended Forecast)	N/A	N/A	Yes
Atmospheric Moisture Profile	VIIRS_GD_11.4.2 NCEP	VIIRS_GD_11.4.2 NCEP (Extended Forecast)	N/A	N/A	Yes
Geopotential Height Profile	VIIRS_GD_11.4.4 NCEP	VIIRS_GD_11.4.4 NCEP (Extended Forecast)	N/A	N/A	Yes
Tropopause Height	VIIRS_GD_09.4.6 NCEP	VIIRS_GD_09.4.6 NCEP (Extended Forecast)	N/A	N/A	Yes
Surface Air Temperature	VIIRS_GD_09.4.10 NCEP	VIIRS_GD_09.4.10 NCEP (Extended Forecast)	N/A	N/A	Yes
Adjusted Surface Pressure	VIIRS_GD_09.4.9 NCEP	VIIRS_GD_09.4.9 NCEP (Extended Forecast)	N/A	N/A	Yes

2.1.3.2 Graceful Degradation Processing

None.

2.1.3.3 Graceful Degradation Outputs

None.

2.1.4 Exception Handling

The ATBD defines error handling as the procedure for handling missing or degraded data or a degraded processing environment. In that document, potential sources of back-up data are identified. These are presented in Table 31. The current algorithm does not include any logic to address these alternative sources of data. Error handling has been added to avoid divide-by-zero errors and to avoid processing with invalid ancillary data.

Table 31. Error Handling for Missing Data

Input	Description	Comments
SDRs	10.763 radiance and brightness temperatures	No secondary source identified.
Cloud Mask	Cloud/no cloud for each pixel	Simple cloud mask could be implemented based on SDR thresholds.
Cloud Phase	Ice or water cloud flag	CMIS IWC and CLW data may be potential backup.
COT	Cloud optical thickness	No secondary source. Algorithm could forego RTM calculations and assign CTT based on brightness temp only.
EPS	Cloud effective particle size	No secondary source. See COT.
CTT	Cloud top temperature as determined by the VIIRS COP Unit	Other NPOESS CTT.
NCEP Atmospheric data	Atmospheric temperature and moisture as function of pressure and height	CMIS or CrIS could be used as secondary backup.

2.1.5 Data Quality Monitoring

Currently, the only outputs from the CT algorithm are the CTT, CTp, and CTH IPs plus the Quality Parameters for each pixel.

Suggested additional QC data are given in Appendix B.

2.1.6 Computational Precision Requirements

Computations carried out in the NonDayWaterCloud() module are performed in double precision in the current version. Single precision should be sufficient for these calculations though tests should be performed to confirm this. All other modules, including the OSS forward model, use single precision.

Note also the CTT, CTp, and CTH products are stored as *floats*. This is sufficient to meet the accuracy and precision requirements on these IPs. The Generate Cloud EDRs (GCE) algorithm produces both the CTp and CTH outputs as scaled *unsigned shorts* (16 bits).

The OSS forward model achieves an increase in computational efficiency through training with respect to a slower Line-By-Line RTM (LBLRTM). This training is performed assuming a particular shape for the spectral response function. In the current implementation, a square response has been assumed for the M15 band. The OSS databases need to be regenerated prior to implementation of the algorithm, once the true response is known.

2.1.7 Algorithm Support Considerations

Some algorithm parameters may require future tuning. Table 31 gives a list of the CTP algorithm parameters with their descriptions, locations in the software, and current assigned values.

Table 32. List of Algorithm Parameters

Note: (parameters varying by sensor marked with an *)

Algorithm Parameter	Assigned Values	Description	Parameter Location
NPLEVELS	26	Number of pressure levels in NCEP data	ctp.h
MAX_SOLN	20	Maximum number of possible solution to non-day/water cloud retrieval	ctp.h
REGRESS_COEF_A1	9.5	Coefficient for calculating virtual temperature	ctp.h
REGRESS_COEF_B1	265.5	Coefficient for calculating virtual temperature	ctp.h
REGRESS_COEF_A2	7.5	Coefficient for calculating virtual temperature	ctp.h
REGRESS_COEF_B2	237.3	Coefficient for calculating virtual temperature	ctp.h
PRESSURE_CONV_FACT	6.1078	Coefficient for calculating virtual temperature	ctp.h
REGRESS_CONST	0.379	Coefficient for calculating virtual temperature	ctp.h
NONDAY_MAXCTH	-1000.0	Max CTH for non-day/water calculations	ctp.h
NONDAY_MINCTP	100000.0	Min CTP for non-day/water calculations	ctp.h
minCtt	175	Minimum temperature for CTT	coefficients input file
maxCtt	310	Maximum temperature for CTT	coefficients input file
minCth	0	Minimum height for CTH	coefficients input file
maxCth	20	Maximum height for CTH	coefficients input file
minCtp	50	Minimum pressure for CTP	coefficients input file
maxCtp	1050	Maximum pressure for CTP	coefficients input file
MXLEV	24	Maximum number of levels used in OSS (for memory allocation)	ctp.h
MXCHAN	100	Maximum number of channels used in OSS (for memory allocation)	ctp.h
maxRTLEVELS	100	Maximum number of levels in RT calculation (for memory allocation)	ctp.h
em_snow	0.988	Snow emissivity	ctp.h
em_ice	0.988	Ice emissivity	ctp.h
em_desert	0.966	Desert emissivity	ctp.h
em_land	0.980	Land emissivity	ctp.h
em_inlandwater	0.991	Inland water emissivity	ctp.h
em_seawater	0.991	Seawater emissivity	ctp.h
em_coastal	0.984	Coastal emissivity	ctp.h
numAggAt	8	Number of pixels along track in analysis block	coefficients input file
numAggXt	8	Number of pixels along scan in analysis block	coefficients input file
maxIterRt	10	Maximum number of allowed iteration of day/water CTP retrieval	coefficients input file
RADNOISE*	0.01	Radiometric noise amplitude in M15 band	ctp.h

Algorithm Parameter	Assigned Values	Description	Parameter Location
chiSqFit	1	Chi-square requirement for convergence	coefficients input file
thkCot1	1	COT threshold for cloud thickness table	coefficients input file
thkCot2	3	COT threshold for cloud thickness table	coefficients input file
thkCtp1	600	CTP threshold for cloud thickness table	coefficients input file
thkCtp2	800	CTP threshold for cloud thickness table	coefficients input file
cldThick[3][3]	{{200,100,50}, {150,75,38}, {100,50,25}}	Cloud thickness as a function of COT and CTP as specified by thkCot1, thkCot2, thkCtp1, and thkCtp2	coefficients input file
pw0	0.067	PW regression coefficient	coefficients input file
pw1	-0.002	PW regression coefficient	coefficients input file
pw2	0.22	PW regression coefficient	coefficients input file
pw3	0.105	PW regression coefficient	coefficients input file
blkCloudCot	200	Default COT for black clouds in WindowIR retrieval	coefficients input file
blkCloudEps	10	Default EPS for black clouds in WindowIR retrieval	coefficients input file
maxVertTemp	325	Maximum vertical temperature	coefficients input file
minVertTemp	180	Minimum vertical temperature	coefficients input file
maxVertWaterVap	10	Maximum vertical water vapor	coefficients input file
minVertWaterVap	1e-09	Minimum vertical water vapor	coefficients input file
minTempProf	100	Minimum valid temperature profile value	coefficients input file
RAD_WL_TO_FREQ*	11.5592169	Radiance units conversion factor applied to OSS calculations	ctp.h
dayThresh	1.39626	Day/night solar zenith angle threshold	coefficients input file

2.1.8 Assumptions and Limitations

2.1.8.1 Assumptions

The CTP retrieval algorithm assumes the VIIRS 750m SDR, the VCM IP including Cloud Phase, the VIIRS COT and EPS IPs, and CTT IP determined by the COP retrieval are all available for processing. The algorithm also requires that atmospheric data including temperature and moisture profiles, from a recent NCEP forecast has been obtained.

The COT IP input is assumed to pertain to a vertical path and apply to a wavelength of 550 nm.

The input NWP height profile is in terms of geopotential height. The gridding design should be reviewed to confirm this assumption. Also, see Appendix A for a discussion of related issues.

The code determines the process path (ProcessDayWaterCloud versus ProcessNonDayWaterCloud) based on the presence of a valid CTT variable as computed by the COP IP. Thus the determination of cloud phase and day versus night is done once in the COP IP Unit and not independently checked in this unit.

2.1.8.2 Limitations

CTP retrievals are performed only when all required inputs are available. No logic is included to replace missing data with secondary sources or to implement a back-up algorithm when certain inputs are not available.

The CTP QC flag (confidence flag) is based solely on the quality of the cloud mask inputs and does not reflect quality assessments of other inputs including CTT for the non-day/water cloud algorithm and SDRs, COT, and EPS for the day/water cloud algorithm. The quality flag does not reflect levels of confidence associated with the retrieval when, for example, more than one solution is identified by the non-day/water algorithm. Recommendations for an additional 16-bit QC mask are given in Appendix B.

The code uses the surface type as determined by the cloud mask. A decision was made in the development of Version 6 to not add any new data dependencies to the cloud module. Improved accuracy for optically thin clouds would likely result from using a product related to the surface type EDR. An analysis would be required to assess which is the best surface type product to employ and its impact on the dataflow and latency. It could be implemented as a future EDR upgrade. Minor code modifications would be required in this case to: (a) read in appropriate surface type; (b) modify the surface emissivity LUT and LUT handling code for the new surface types.

For day/water conditions, the code requires both COT and EPS inputs. Since during glint conditions, these quantities are not computed by the COP IP Unit, this code does not produce the CTP_IP outputs in glint conditions. This limitation may be addressed in a future “delta’ release of the code.

2.1.8.3 Code Migration Notes

The next version of the VCM is expected to add another “cloud phase” indicator: multi-layer ice over water cloud. The algorithm logic needs to be reviewed to ensure it will work with this change.

The text has noted those files and configuration parameters that will vary by sensor and/ or may need to be updated.
They are summarized here:

- Parameters dependent on sensor bandpass contained in ctp.h (those are marked with an * in Table 31)
- All input algorithm parameter files associated with the day/water algorithm (see Table 9)

Improved performance is expected to result from using an NWP product surface pressure rather than relying on interpolation of the height profile. This data is produced by NCEP and other central sites and is used by other EDRs. Some minor changes may be required to read this value from the ancillary inputs. The algorithm currently looks for this value in the surface temperature file and uses it if present.

The algorithm uses the surface air temperature NWP data. It would be preferable to use NWP skin temperature in its place. The skin temperature is produced by most current NWP models.

3.0 GLOSSARY/ACRONYM LIST

3.1 Glossary

The current glossary for the NPOESS program, D35836_G_NPOESS_Glossary, can be found on eRooms. Table 33 contains those terms most applicable for this OAD.

Table 33. Glossary

Term	Description
Algorithm	A formula or set of steps for solving a particular problem. Algorithms can be expressed in any language, from natural languages like English to mathematical expressions to programming languages like FORTRAN. On NPOESS, an algorithm consists of: 1. A theoretical description (i.e., science/mathematical basis) 2. A computer implementation description (i.e., method of solution) 3. A computer implementation (i.e., code)
Algorithm Configuration Control Board (ACCB)	Interdisciplinary team of scientific and engineering personnel responsible for the approval and disposition of algorithm acceptance, verification, development and testing transitions. Chaired by the Algorithm Implementation Process Lead, members include representatives from IWPTB, Systems Engineering & Integration IPT, System Test IPT, and IDPS IPT.
Algorithm Verification	Science-grade software delivered by an algorithm provider is verified for compliance with data quality and timeliness requirements by Algorithm Team science personnel. This activity is nominally performed at the IWPTB facility. Delivered code is executed on compatible IWPTB computing platforms. Minor hosting modifications may be made to allow code execution. Optionally, verification may be performed at the Algorithm Provider's facility if warranted due to technical, schedule or cost considerations.
Ancillary Data	Any data which is not produced by the NPOESS System, but which is acquired from external providers and used by the NPOESS system in the production of NPOESS data products.
Auxiliary Data	Auxiliary Data is defined as data, other than data included in the sensor application packets, which is produced internally by the NPOESS system, and used to produce the NPOESS deliverable data products.
EDR Algorithm	Scientific description and corresponding software and test data necessary to produce one or more environmental data records. The scientific computational basis for the production of each data record is described in an ATBD. At a minimum, implemented software is science-grade and includes test data demonstrating data quality compliance.
Environmental Data Record (EDR)	*[IORD Definition]* Data record produced when an algorithm is used to convert Raw Data Records (RDRs) to geophysical parameters (including ancillary parameters, e.g., cloud clear radiation, etc.). *[Supplementary Definition]* An Environmental Data Record (EDR) represents the state of the environment, and the related information needed to access and understand the record. Specifically, it is a set of related data items that describe one or more related estimated environmental parameters over a limited time-space range. The parameters are located by time and Earth coordinates. EDRs may have been resampled if they are created from multiple data sources with different sampling patterns. An EDR is created from one or more NPOESS SDRs or EDRs, plus ancillary environmental data provided by others. EDR metadata contains references to its processing history, spatial and temporal coverage, and quality.
Model Validation	The process of determining the degree to which a model is an accurate representation of the real-world from the perspective of the intended uses of the model. [Ref.: DoDD 5000.59-DoD Modeling and Simulation Management]
Model Verification	The process of determining that a model implementation accurately represents the developer's conceptual description and specifications. [Ref.: DoDD 5000.59-DoD Modeling and Simulation Management]
Operational Code	Verified science-grade software, delivered by an algorithm provider and verified by IWPTB, is developed into operational-grade code by the IDPS IPT.

Term	Description
Operational-Grade Software	Code that produces data records compliant with the System Specification requirements for data quality and IDPS timeliness and operational infrastructure. The software is modular relative to the IDPS infrastructure and compliant with IDPS application programming interfaces (APIs) as specified for TDR/SDR or EDR code.
Raw Data Record (RDR)	*[IORD Definition]* Full resolution digital sensor data, time referenced and earth located, with absolute radiometric and geometric calibration coefficients appended, but not applied, to the data. Aggregates (sums or weighted averages) of detector samples are considered to be full resolution data if the aggregation is normally performed to meet resolution and other requirements. Sensor data shall be unprocessed with the following exceptions: time delay and integration (TDI), detector array non-uniformity correction (i.e., offset and responsivity equalization), and data compression are allowed. Lossy data compression is allowed only if the total measurement error is dominated by error sources other than the data compression algorithm. All calibration data will be retained and communicated to the ground without lossy compression. *[Supplementary Definition]* A Raw Data Record (RDR) is a logical grouping of raw data output by a sensor, and related information needed to process the record into an SDR or TDR. Specifically, it is a set of unmodified raw data (mission and housekeeping) produced by a sensor suite, one sensor, or a reasonable subset of a sensor (e.g., channel or channel group), over a specified, limited time range. Along with the sensor data, the RDR includes auxiliary data from other portions of NPOESS (space or ground) needed to recreate the sensor measurement, to correct the measurement for known distortions, and to locate the measurement in time and space, through subsequent processing. Metadata is associated with the sensor and auxiliary data to permit its effective use.
Retrieval Algorithm	A science-based algorithm used to 'retrieve' a set of environmental/geophysical parameters (EDR) from calibrated and geolocated sensor data (SDR). Synonym for EDR processing.
Science Algorithm	The theoretical description and a corresponding software implementation needed to produce an NPP/NPOESS data product (TDR, SDR or EDR). The former is described in an ATBD. The latter is typically developed for a research setting and characterized as "science-grade".
Science Algorithm Provider	Organization responsible for development and/or delivery of TDR/SDR or EDR algorithms associated with a given sensor.
Science-Grade Software	Code that produces data records in accordance with the science algorithm data quality requirements. This code, typically, has no software requirements for implementation language, targeted operating system, modularity, input and output data format or any other design discipline or assumed infrastructure.
SDR/TDR Algorithm	Scientific description and corresponding software and test data necessary to produce a Temperature Data Record and/or Sensor Data Record given a sensor's Raw Data Record. The scientific computational basis for the production of each data record is described in an Algorithm Theoretical Basis Document (ATBD). At a minimum, implemented software is science-grade and includes test data demonstrating data quality compliance.
Sensor Data Record (SDR)	*[IORD Definition]* Data record produced when an algorithm is used to convert Raw Data Records (RDRs) to calibrated brightness temperatures with associated ephemeris data. The existence of the SDRs provides reversible data tracking back from the EDRs to the Raw data. *[Supplementary Definition]* A Sensor Data Record (SDR) is the recreated input to a sensor, and the related information needed to access and understand the record. Specifically, it is a set of incident flux estimates made by a sensor, over a limited time interval, with annotations that permit its effective use. The environmental flux estimates at the sensor aperture are corrected for sensor effects. The estimates are reported in physically meaningful units, usually in terms of an angular or spatial and temporal distribution at the sensor location, as a function of spectrum, polarization, or delay, and always at full resolution. When meaningful, the flux is also associated with the point on the Earth geoid from which it apparently originated. Also, when meaningful, the sensor flux is converted to an equivalent top-of-atmosphere (TOA) brightness. The associated metadata includes a record of the processing and sources from which the SDR was created, and other information needed to understand the data.

Term	Description
Temperature Data Record (TDR)	*[IORD Definition]* Temperature Data Records (TDRs) are geolocated, antenna temperatures with all relevant calibration data counts and ephemeris data to revert from T-sub-a into counts. *[Supplementary Definition]* A Temperature Data Record (TDR) is the brightness temperature value measured by a microwave sensor, and the related information needed to access and understand the record. Specifically, it is a set of the corrected radiometric measurements made by an imaging microwave sensor, over a limited time range, with annotation that permits its effective use. A TDR is a partially-processed variant of an SDR. Instead of reporting the estimated microwave flux from a specified direction, it reports the observed antenna brightness temperature in that direction.

3.2 Acronyms

The current acronym list for the NPOESS program, D35838_G_NPOESS_Acronyms, can be found on eRooms. Table 34 contains those terms most applicable for this OAD.

Table 34. Acronyms

Acronym	Description
AM&S	Algorithms, Models & Simulations
API	Application Programming Interfaces
ARP	Application Related Product
CBH	Cloud Base Height
CCL	Cloud Cover Layers
CDA	Command and Data Acquisition
CDFCB-X	Common Data Format Control Book - External
CI	Configured Item
COP	Cloud Optical Properties
COT	Cloud Optical Thickness
CT	Cloud Top
CTH	Cloud Top Height
CTP	Cloud Top Parameters
CTp	Cloud Top Pressure
CTT	Cloud Top Temperature
DDD	Detailed Design Document
DMS	Data Management Subsystem
DPIS ICD	Data Processor Inter-subsystem Interface Control Document
DQTT	Data Quality Test Table
EPS	(Cloud) Effective Particle Size
GCE	Generate Cloud EDRs
IIS	Intelligence and Information Systems
INF	Infrastructure
ING	Ingest
IP	Intermediate Product
LBLRTM	Line-by-line Radiative Transfer Model
LUT	Look-Up Table or Local User Terminal
MDFCB	Mission Data Format Control Book
MSC	Multiple Scattering Correction
OSS	Optimal Spectral Sampling
PPC	Perform Parallax Correction
PRO	Processing
QF	Quality Flag
RTM	Radiative Transfer Model
SDR	Sensor Data Records
SI	International System of Units
TBD	To Be Determined
TBR	To Be Resolved
TOA	Top of the Atmosphere
UCLA	University of California, Los Angeles
USB	Unified S-band
VCM	VIIRS Cloud Mask

4.0 OPEN ISSUES

Table 35. TBXs

TBX ID	Description	Resolution Date
None		

Appendix A. Definition of Cloud Top Height

The CTP IP software outputs CTH based on geopotential height above the local geoid as illustrated in Figure 7.

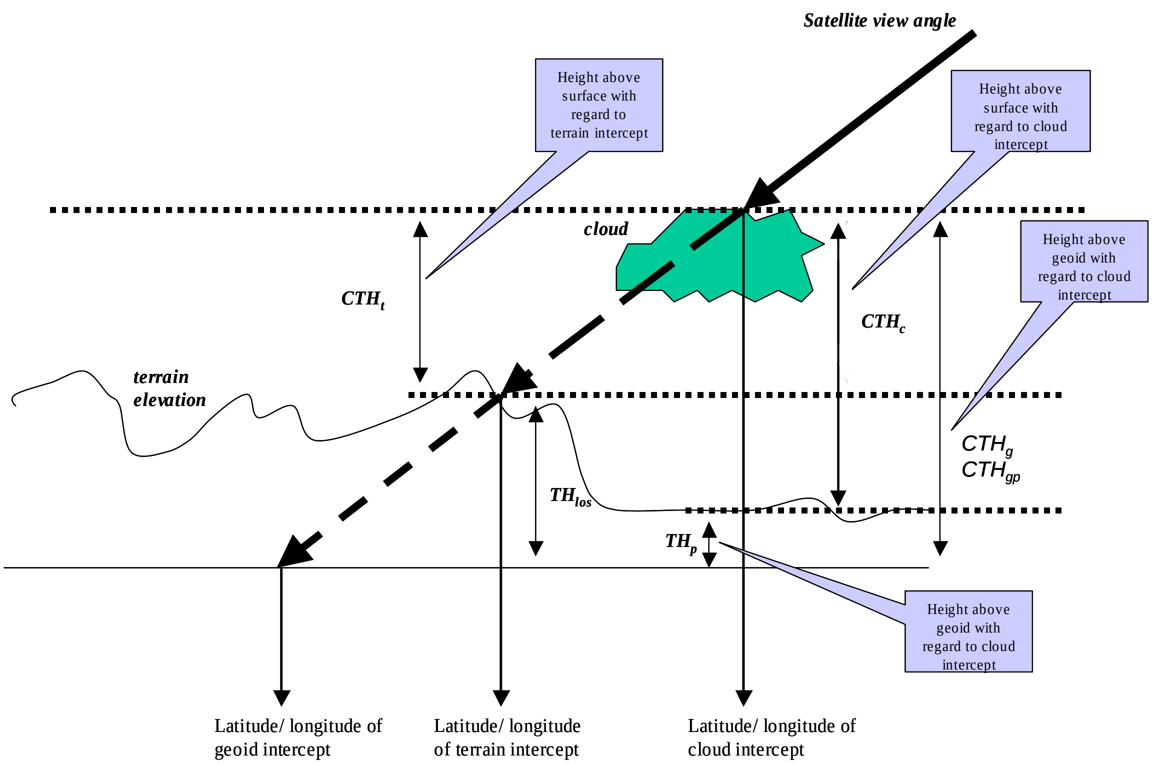

Figure 7. Illustration of Cloud Top Geometry

The CTP algorithm assumes the NWP Height Profile is in terms of geopotential height as this is the standard at the various forecasting centers. This assumption should be verified. The algorithm derives the cloud height using this height profile input and so it is defined also in terms of geopotential height[1].

The System Specification defines the cloud EDRs in terms of height above the terrain immediately below the cloud, the quantity CTH_c in the figure. This is computed simply as:

$CTH_c = (g_l / g_0)\, CTH_{cp} - TH_p$

Where g_l is the local acceleration of gravity (at the given altitude and latitude)
g_0 is the reference acceleration of gravity for the geopotential height calculation and the other terms are as defined in the figure.

This calculation can be easily performed in the parallax correction software unit.

[1] Since the geoid and acceleration of gravity vary very slowly over the scales of interest, the quantities CTH_g, CTH_{gp} differ negligibly for definitions based on the cloud, terrain or geoid intercepts.

Appendix B. Recommended Additional Quality Flags and Diagnostics

Version 6 of the code was design to produce the same output and quality flags as Version 5. Version 5 calculated a confidence flag based solely on the cloud mask confidence. It included no additional information on the algorithm.

Additional consideration has led to the recommendation for adding some additional quality flags and diagnostic variables. Since IDPS sizing has been performed without these additional variables, the IDPS impact should be considered. Only minor code modifications would be required to implement these flags if they are adopted.

These recommendations for the added flags and diagnostic information are being reviewed by the NPOESS SE M&S group. The changes, if adopted, would be implemented by the NPOESS SE M&S Group and included in a revised algorithm drop at a future date.

The data below would be added to the IP output file. These variables would be used for algorithm Cal/ Val, tuning, trouble-shooting and maintenance. They would only be useful if access to the CTP IP product is provided, so this recommendation is being reviewed as part of the larger discussion of IP distribution.

The output data are of two types:

- Quality Flags: Integral data including yes/ no, on/ off masks and numerical categorical data small in size (typically total flags are one to a few bytes per pixel)
- Diagnostic data numerical data (ints, doubles, floats, arrays) characterizing quantitative performance of the algorithm or specific algorithm steps

We recommend the quality flags defined in Table 36 be produced at all times.

The diagnostic data are only needed for debugging purposes, algorithm tuning/ trouble shooting or Cal/ Val. The principal information desired for trouble shooting from the day/water algorithm is information on the output of each iteration. The following diagnostic data are recommended:

- Independent of iteration
    - Model noise value
- Last iteration
    - Radiance derivative
    - Local Lapse rate (K/ km)
- At least last five iterations, preferably all
    - Computed CTP and Chi-sq

The ability of HDF-5 to store ragged arrays would enable all iterations to be stored without unnecessary wasted space. It is assumed this information would be saved for limited, selected regions.

Table 36. CTP Quality Flag Recommended Content

Note: (DWC = day/water cloud; NDWC = non-day/water cloud); when the pixel was not processed by the CTP algorithm, e.g., not a cloud, then all bits = 0)

Bit(s)	Desc.	Content
0	Valid IP processed this pixel	0 = CTP IP algorithm not run on this pixel 1 = CTP IP attempted to process this pixel
1	Valid ancillary data	0 = some of required ancillary data missing or invalid 1 = all required ancillary data present
2	Valid SDR data	0 = missing required sensor data 1 = all required data are present (note this is only tested for day/ water)
3	Valid COP IP data	0 = missing or invalid required intermediate product from COP unit 1 = required outputs from COP IP unit present
4	Valid auxiliary data	0 = missing or invalid AUX data 1 = all required AUX data present
5-6	Algorithm converged	DWC: 00 = did not converge to useful Chi-sq value 01 = poor quality convergence, but might be useful 10 = approximately meets convergence criteria 11 = much better than convergence criteria NDWC: Always 00
7	First guess source	DWC: 0 = brightness temperature 1 = nearby pixel NWDC: Always 0
8-12	Number of iterations or solutions	DWC: 0 to 32 iterations (31 means 31 or greater) NDWC: Number of solutions identified for search (31 means 31 or greater)
13-14	Cloud phase	DWC and NDWC: 00 = ice phase 01 = water phase 10 = mixed phase 11 = overlapping ice/ water (future)
15	Day/ night flag	DWC and NDWC: 0 = day